

# **A Visual Teaching and Learning Environment for the Open Shortest Path First Routing Protocol**

**Eric Gamess**

Florida Atlantic University, Boca Raton, Florida, USA, egamess@fau.edu.  
Universidad Central de Venezuela, Caracas, Venezuela.

**Jesús Urbáez**

Universidad Central de Venezuela, Caracas, Venezuela, jesusurbaz@gmail.com.

**Arturo Palacios**

Universidad Central de Venezuela, Caracas, Venezuela, arturo7200@gmail.com.

## **ABSTRACT**

In this paper we present Easy-OSPF, an implementation of the Open Shortest Path First (OSPF) protocol developed in C#. Its main goal is to be used as a didactic application in introductory and advanced network courses to support the teaching and learning of OSPF. It allows clarifying the complex and abstract theory of OSPF through virtual laboratories using graphics, images, and animations, which can be easily transformed in knowledge. Easy-OSPF offers a user-friendly, flexible and robust graphical user interface that allows users to configure all relevant aspects of the network and the OSPF protocol. It also provides a network viewer, an integrated sniffer, and a protocol table viewer. The initial teaching experiments that we conducted with the usage of Easy-OSPF indicate that it can dramatically facilitate the learning of OSPF. Additionally, Easy-OSPF can be used in a production network as a functional but limited implementation of the OSPF protocol.

**Keywords:** OSPF, Routing Protocols, Visual Learning, Didactic Applications, Virtual Laboratories.

## **1. INTRODUCTION**

Visual learning is the use of graphics, images, and animations to enable and enhance learning. It is a proven method of education in which ideas, concepts, data, and other information are associated with images and animations, resulting in an easier and more effective method of transmitting skills. Students can understand theoretical concepts much easier if they can see them, or interact with them as in the real life. Visual learning uses methods that help students to open their minds and think graphically.

For these reasons, the Graphical User Interface (GUI) is one of the most important parts of any didactic tool. It is the boundary between the application and users. It can be seen, it can be heard, and it can be touched. The piles of software code are invisible, hidden behind the screen, speaker, keyboard, and mouse. According to (Galitz, 2007), the goal of interface design is to make the working with a computer easy, productive, and enjoyable. These characteristics present in a teaching and learning application makes it extremely powerful and efficient to bring knowledge to end-users.

As networking systems are becoming more complex, new curricula and learning tools are needed to help students to acquire solid skills about the complex networking technology. In this paper, we introduce Easy-OSPF, a visual teaching and learning environment for the Open Shortest Path First (OSPF) protocol, and we report how this application can enhance, ease, and make more natural the knowledge and skills transfer.

The rest of the paper is organized as follows. In Section 2, we discuss the fundamental concepts of OSPF. In Section 3, we introduce our new visual teaching and learning environment. Section 4 presents some initial results

of teaching and learning experiments conducted on a group of students using our environment. Related works are presented in Section 5, and finally we conclude the paper in Section 6.

## 2. AN OVERVIEW OF THE OPEN SHORTEST PATH FIRST ROUTING PROTOCOL

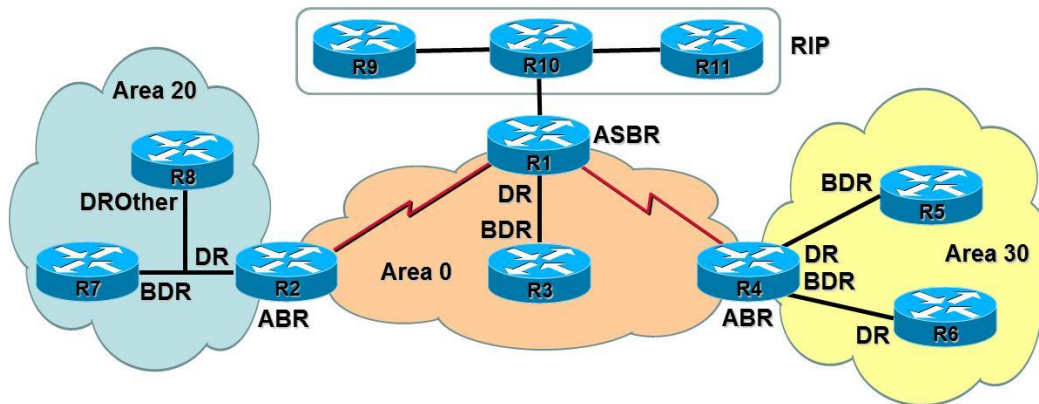
Interior routing protocols can be divided in “distance vector” and “link state” protocols. Distance vector routing protocols, such as the Routing Information Protocol (RIP) (Doyle and Carroll, 2005), broadcast (RIPv1) or multicast (RIPv2) their complete routing table periodically, regardless of whether the routing table has changed. This periodic advertisement is done every 30 seconds in the case of RIP. Each recipient of this information adds a distance vector (its own distance value to the transmitter) to the received information before it forwards it to its neighbors. When the network is stable, distance vector protocols waste a lot of bandwidth because of the periodic sending of routing table updates without any changes. When a failure occurs in the network, they take a long time to reconverge to an alternate path or to flush a dead path from the routing table.

Link state routing protocols were designed to address the limitations of distance vector routing protocols (slow convergence, unnecessary bandwidth usage, and routing loops). Link state protocols are more complex than distance vector protocols, and running them adds an important overhead to the router in the form of memory and CPU utilization. Link state protocols minimize bandwidth usage by sending updates only when a change occurs. A hello mechanism ascertains reachability of neighbors. When a failure occurs in the network, link state protocols flood Link-State Advertisements (LSAs) throughout an area. Using these advertisements, each router creates a database detailing the current network topology. The topology database in each router of the same area is identical, and new LSAs cause every router within the failed area to recalculate routes. Because of their complexity and the flooding process, link state protocols have a poor scalability. As a countermeasure of the scalability problem, the entire network can be divided in area, to limit both the size of the topology database and the flooding of the LSAs, and hence, to reduce the recalculation of the shortest routes to the different destination of the network.

Open Shortest Path First (OSPF) is one of the most commonly-used link state routing protocols. It is standardized by the Internet Engineering Task Force and its current version (version 2 in the case of IPv4) is presented in (Moy, 1998). In fact, (Baker, 1995) states that OSPF is the only required dynamic routing protocol for IPv4 routers. OSPF offers equal cost load balancing, logical partitioning of the network in areas, aggregation (also known as summarization) of routing announcements (to reduce the advertisement of unnecessary subnet information), support for authentication (plain-text and MD5), and support for Variable Length Subnet Mask (VLSM) and Classless Inter-Domain Routing (CIDR). Figure 1 shows an OSPF network with multiple areas. Area 0, also known as the backbone area, must always be present. Other areas (in Figure 1, area 20 and area 30) must be connected to the backbone area. OSPF offers a way to overcome this design limitation through virtual links. OSPF defines several types of routers (intra-area routers, area border routers, and autonomous system boundary routers). Intra-area routers are located entirely within an OSPF area. In Figure 1, R7 and R8 are intra-area routers of area 20. Intra-area routers maintain one topology database for their local area, so in general they do not need to have much memory and CPU power. Area Border Routers (ABRs) are used to connect non-backbone areas to the backbone. In Figure 1, R2 and R4 are ABRs since they connect area 20 to the backbone and area 30 to the backbone, respectively. ABRs maintain a separate topology database for each attached area. Autonomous System Boundary Routers (ASBRs) are located at the boundary between an OSPF network and a non-OSPF network (e.g., RIP, EIGRP, IS-IS). They function as a gateway exchanging routing information between the OSPF network and other routing environments. In Figure 1, R1 is an ASBR and connects the OSPF network to a RIP network.

The exchange of LSAs between neighbors can create significant overhead of network traffic in Broadcast Multi-Access (BMA) networks. To reduce the total bandwidth required to synchronize databases, OSPF introduced the notion of Designated Routers (DRs) and Backup Designated Routers (BDRs). In each BMA network, a DR and a BDR are elected. The other routers of the BMA network are called DROthers. The DR is the central point to exchange LSAs, and the BDR assumes DR functionality when the DR fails. To do so, the DR and BDR form adjacencies with all routers on the BMA network. Each interface of a router is assigned an 8-bit priority (called

router priority), indicating its ability to be selected as DR or BDR. A router priority of zero indicates that the router is not eligible to be selected as DR or BDR. A higher value of the priority represents a better chance to be elected as DR or BDR. However, the election is not preemptive. In Figure 1, R2 is the DR of the BMA network formed by the Ethernet network than connect together R2, R7, and R8. R7, the BDR, monitors the operation of R2 and will become the DR if R2 fails. The other routers, in this case R8, are the DROthers. In OSPF, the DROther with the higher priority will be elected BDR in case of failure of the BDR.



**Figure 1: Example of OSPF Network with Multiple Areas**

OSPF is based on the Shortest-Path First (SPF) algorithm of Dijkstra. In an OSPF network, routers obtain the entire topology database (i.e., all the LSAs) through reliable flooding. Then, each router runs the SPF algorithm on its database to build the IP routing table. OSPF has several types of LSAs with allow the description of the entire OSPF network and external environments: Router LSAs, Network LSAs, Summary LSAs, ASBR-Summary LSAs, External LSAs, and NSSA-External (Not-So-Stubby Area) LSAs. A Router LSA (Type-1) is generated by each OSPF router and describes the state of the router's interfaces within the area. A Network LSA (Type-2) is generated by the DR of each BMA network and lists the routers connected to this network. Summary LSAs (Type-3) are generated by ABRs and represent networks learnt from an area and introduced into another area. ASBR-Summary LSAs (Type-4) are generated by ABRs and describe routes to ASBRs. External LSAs (Type-5) are created by ASBRs and describe routes to destinations that are outside the OSPF network. NSSA-External LSAs (Type-7) are used by routers to send external routing information within a Not-So-Stubby Area (NSSA). The ABRs of the NSSA then translate the Type-7 LSAs to Type-5 LSAs (External LSAs) and flood them as normal Type-5 LSAs to the rest of the OSPF network.

OSPFv2 packets are transmitted in IPv4 datagrams with a protocol identifier equals to 89. That is, OSPF does not use TCP or UDP in the transport layer. OSPF uses multicast and has two assigned multicast IPv4 addresses: 224.0.0.5 (also known as AllSPFRouters) which allows the sending of packets to all the OSPF routers of a local network, and 224.0.0.6 (also called AllDRouters) which serves for the communication with the DR and the BDR of a local BMA network. OSPF packets have a common header plus a specific part, resulting in five different types of packets: Hello (Type-1), Database Description (Type-2), Link-State Request (Type-3), Link-State Update (Type-4), and Link-State Acknowledgment (Type-5).

- Hello: These packets are used as a keepalive mechanism to discover and maintain neighbor relationships.
- Database Description (DBD): When a router establishes an adjacency with another router, they exchange their knowledge of the topology database (set of LSAs) through Database Description packets. In fact, these packets do not carry the whole LSAs, but a summary of the LSAs which can be obtained with a Link-State Request packet.
- Link-State Request (LSR): These packets are seen in the establishment of an adjacency to require missing LSAs. Also, when a router detects that some of its LSAs are out-of-date, it sends a Link-State Request packet to a neighbor requesting them.

- Link-State Update (LSU): Are sent in response to Link-State Request packets, or to propagate a new or more updated version of a LSA in the OSPF network. A Link-State Update can carry one or more LSAs. They are acknowledged with Link-State Acknowledge packets and resent if a retransmission is necessary.
- Link-State Acknowledgment (LSAck): A router sends this type of packets in response to Link-State Update packets to inform that the LSAs have been received successfully.

As it can be easily inferred from the previous short overview of the OSPF operation, it is not easy to teach or learn these complex concepts. Therefore, we decided to develop a graphic oriented application to support introductory and advanced courses for the training of network specialists.

### 3. EASY-OSPF

Easy-OSPF is an implementation of the OSPF protocol developed in C# (Skeet, 2010). C# is a simple, modern, general-purpose, powerful, object-oriented language originally developed by Microsoft within the .NET initiative and later approved as a standard by the International Organization for Standardization (ISO/IEC 23270). We chose C# since it easily allows the writing of complex applications and it is available in all common operating systems (Windows, Unix, Mac OS X) through the Mono Project<sup>1</sup> for free. We also used the following libraries in the development:

- SharpPcap<sup>2</sup>: Is a packet capture framework for the .NET environment based on WinPcap, a famous industry-standard tool for link-layer network access in Windows environments. The purpose of this library is to provide an API for capturing, injecting, analyzing and building packets using any .NET language such as C# and VB.NET.
- Pcap.Net<sup>3</sup>: Is a .NET wrapper for WinPcap written in C++ and C#. It has almost all WinPcap features and also includes a packet interpretation framework.
- Piccolo2D<sup>4</sup>: Is a free open source toolkit that supports the development of 2D structured graphics and Zoomable User Interfaces (ZUIs). There are several version of Piccolo2D, and the one that we used (Piccolo2D.NET) is a layer built on top of the .NET Framework. We chose Piccolo2D for its advanced support of zooms. Therefore, when the network is complex, users start having a general view of the network and they can just zoom in for more specific details on a particular region.

Easy-OSPF can be installed in a PC with several Network Interface Cards (NICs) to transform it in an OSPF router. It is a true, but limited, implementation of the OSPF protocol and its main goal is to be used as a didactic application in introductory and advanced network courses to support the teaching and learning of OSPF. The current version of Easy-OSPF provides a set of four modules (Configuration, Network Map, OSPF Tables, and OSPF Sniffer) briefly described below.

#### 3.1 CONFIGURATION MODULE

This module offers users an interface for setting and configuring the network attributes of the PC as well as the environment variables of OSPF. Figure 2 shows how the module is divided in three main sections: one that lists all the NICs detected in the PC (in Figure 2, the PC has three NICs), another section for setting network interfaces attributes (e.g., IP address, network mask, metric, hello interval, dead interval, interface mode, router priority, etc.), and a third one for configuring router parameters (e.g., Router ID and Area ID). This module offers images, icons, and warning messages that guide users in the settings, preventing them from typing incorrect values. That is, if users type an incorrect network mask or a hello interval that is greater than a dead interval, they will be warning about the problem. This graphic oriented configuration is by far much easier and attractive than specifying the setting through the traditional Command-Line Interface (CLI) where users will have to memorize a large amount of commands to configure devices, and therefore, it is more adequate for novice students. In the

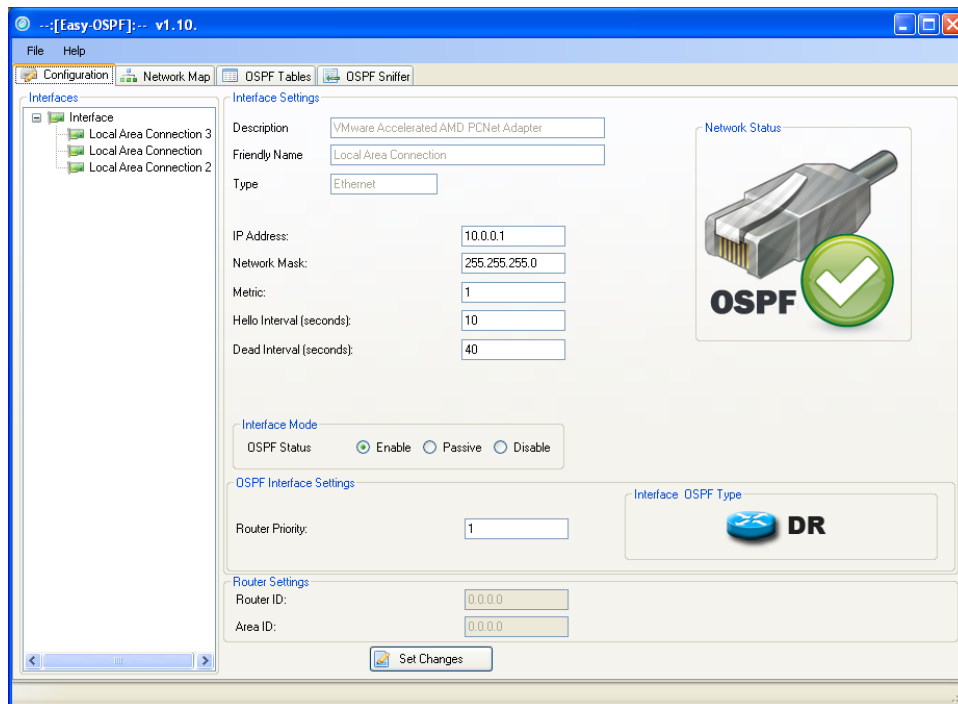
<sup>1</sup> <http://www.mono-project.com>

<sup>2</sup> <http://sourceforge.net/projects/sharppcap>

<sup>3</sup> <http://pcapdotnet.codeplex.com>

<sup>4</sup> <http://www.piccolo2d.org>

“OSPF Interface Type”, users can see graphically the type of the router for a BMA network (DR, BDR, or DROther), and this information will be automatically and dynamically updated when changes occur.



**Figure 2: The Configuration Module of Easy-OSPF**

### 3.2 NETWORK MAP MODULE

The Network Map is a powerful module where users have a dynamic view of the network. The map is generated according to the LSAs that are present in the topology database. This module is very helpful for students since they can see in a graphical way all the topology, i.e., the directly connected neighbors (1-hop neighbors), neighbors that are 2 or more hops away, as well as the interconnection network between these OSPF routers. Professors will find the module especially suitable for teaching the OSPF's metric, which is based on bandwidth and the Dijkstra's algorithm, since the metric of each link appears over the link in the map, allowing an easy way to find the shortest path.

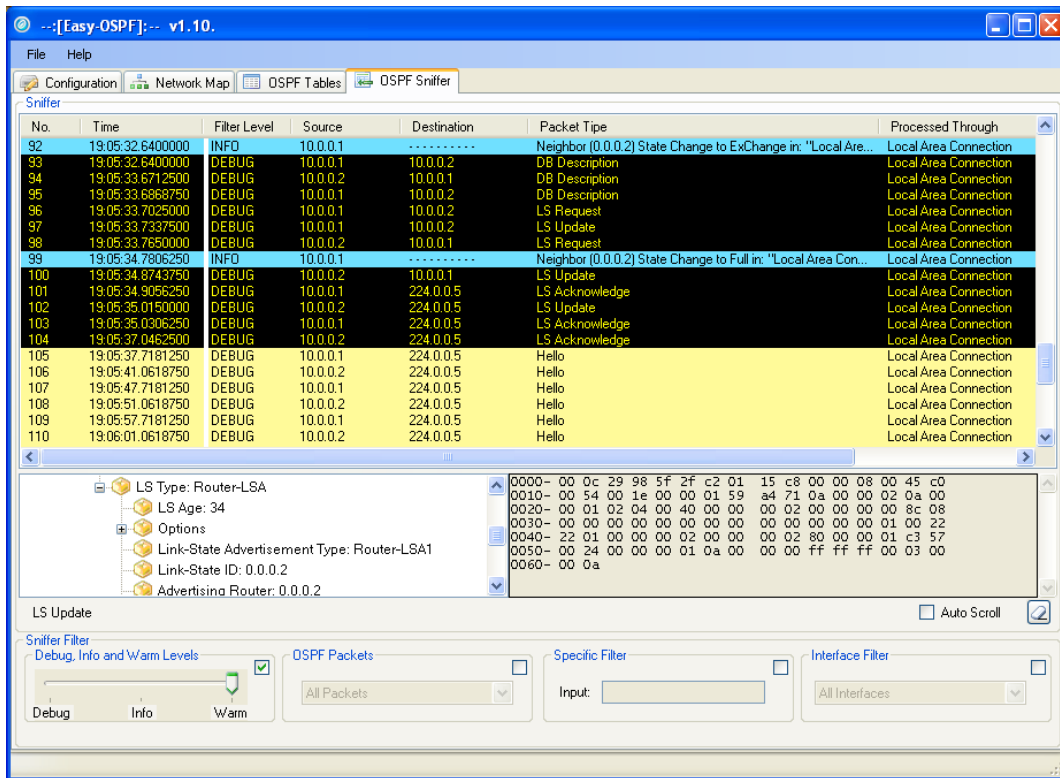
### 3.3 OSPF TABLES MODULE

OSPF handles three main tables (IP Routing Table, Neighbor Table, and Topology Table) that are fundamental for the functionality to the entire protocol. These tables are available in this module and they maintain the same format used by the Cisco CLI, to facilitate users' migration. However, in the case of Easy-OSPF, these tables are updated in real time with the topology changes. It is clear that for the visualization of the tables, users do not have to remember or enter any command, resulting easier for novices students.

### 3.4 OSPF SNIFFER MODULE

The OSPF Sniffer Module (Figure 3) is a complete and powerful debugging module that keeps track of every packet exchange for troubleshooting and learning purposes. Similar to a packet sniffer, users can see all the information (fields with explanations) of the OSPF packets. In addition, system debugging messages are also added by the module. There are three levels of debugging messages: Debug, Info and Warn. The Debug level messages show normal OSPF packets exchanged (Hello, Database Description, Link-State Update, Link-State Request, and Link-State Acknowledgment). The Info level messages provide general information about OSPF's changes (e.g., topology database) and information about the network interfaces (e.g., interfaces going up or down,

enabling or disabling OSPF on interfaces). Finally, the Warn level messages are related to route loss, unexpected network interface disablement, OSPF's packet drop, etc. Additionally, the logger has some Access Control List (ACL) options allowing filtering by packet types, by user custom inputs, or by network interface cards.



**Figure 3: The OSPF Sniffer Module of Easy-OSPF**

### 3.5 VALIDATION OF EASY-OSPF

To validate our implementation of OSPF, we created numerous testbeds where we connected different types of OSPF routers (real Cisco 2811 routers, Zebra<sup>5</sup>, Quagga<sup>6</sup>, XORP, and Vyatta<sup>7</sup>) with Easy-OSPF and made many experiments. We also did several experiments with topologies restricted to Easy-OSPF routers. Finally, we verified that Easy-OSPF could be used in all the practices of the CCNA (Semester 2) and most of the CCNP ROUTE certification program.

### 3.6 LIMITATION OF EASY-OSPF

The current version of Easy-OSPF is limited to a unique area. However, we are currently working to overcome this limitation and we will soon have multi-areas version.

## 4. RESULTS OF INITIAL TEACHING AND LEARNING EXPERIMENTS WITH EASY-OSPF

An exploratory study of Easy-OSPF was conducted to evaluate the effectiveness of this tool in a group of 40 students with different knowledge in terms of networking concepts, ranging from no knowledge at all to advanced skills. The study was carried out in the School of Computer Science of our University (Universidad Central de Venezuela, Caracas, Venezuela). The exercises consisted in setting up some simple topologies with: (1) Packet Tracer, (2) IMUNES, (3) GNS3, and (4) Easy-OSPF. In the case of Easy-OSPF, we created a virtual machine for

<sup>5</sup> <http://www.zebra.org>  
<sup>6</sup> <http://www.quagga.net>  
<sup>7</sup> <http://www.vyatta.com>

each virtual router to run the complete experiments in a unique PC. We used VMWare Player<sup>8</sup>, a famous virtual machine player that can be downloaded for free. The first exercise consisted in understanding the Hello protocol to discover/rediscover the OSPF neighbors. The goal of the second exercise was to develop familiarity with the different LSAs of the OSPF topology database and their relationship with the actual network. The third exercise focused in understanding the different OSPF packets, their objectives, and their structures. In this exercise, students had to identify the different fields of the common header and the specific fields of each packet type (Hello, Database Description, Link-State Request, Link-State Update, and Link-State Acknowledgement). Finally, the objective of the fourth exercise was around authentication and security. In the proposed topology, students had to add a rogue router and evaluate the damaged done by it after propagating fake information in the OSPF network. Students also studied the weaknesses of clear-text authentication.

After doing the experiments, the students filled an anonymous survey to assess certain features of the tools like usability, ease of use and configuration, completeness, application feedback, learning and understanding of theoretical principles of OSPF, traffic analyzer, etc., in a range of 1-5 marks, where 1 mark was the lowest score acceptance and 5 marks the highest. The analysis of the survey gave quite positive results (see Table 1) indicating that Easy-OSPF is well accepted by students and greatly facilitates the teaching and learning of OSPF.

**Table 1: Evaluation of Different Didactic Applications for the Understanding of OSPF**

	<b>Packet Tracer</b>	<b>IMUNES</b>	<b>GNS3</b>	<b>Easy-OSPF</b>
<b>Usability</b>	4.5	4.1	4.2	4.5
<b>Ease to Configure</b>	3.9	3.7	3.8	4.2
<b>Completeness</b>	3.5	4.6	4.7	4.5
<b>Application Feedback</b>	3.1	3.0	2.9	4.2
<b>Learning and Understanding</b>	4.0	3.9	3.9	4.5
<b>Traffic Analyzer</b>	2.5	4.2	4.4	4.7

## 5. RELATED WORKS

The Cisco Networking Academy Program has been very successful all over the world. This program is currently present in high schools, colleges, universities, technical and military schools, community-based and government organizations. Cisco offers a basic course of routing protocols (CCNA semester 2) (Lammle, 2011), as well as advanced courses such as the CCNP ROUTE (Odom, 2010) and the CCIE Routing and Switching (Manning, 2010). In all these courses, students learn OSPF at different levels. To do so, Cisco offers two practical methods, one based on real laboratories and the other on virtual laboratories. In the real laboratories, students use real routers to practice the skills and knowledge viewed in class. In this case, the connection and iteration with the routers are done through a terminal emulator, telnet, or a SSH program using the Command-Line Interface (CLI). The CLI style is powerful, and it is the preferred method used by expert administrators to configure Cisco's devices. However, it requires students to type commands and see the results in plain text. Therefore, the CLI is not very useful to teach routing concepts to novice students since (1) they will have to remember the commands, (2) the syntax of commands can be complex, and (3) it is not possible to visualize dynamic changes in real-time (e.g., the discovery/recovery of neighbors). It is also very prone to typing errors that can lead to novice users' frustration, resulting in an application with poor usability and pedagogy to teach and learn. Last, but not least, it is not always possible to do practices with real routers for reasons of costs.

On the other hand, Cisco offers Packet Tracer for free to students of the Cisco Networking Academy Program. Packet Tracer is a powerful didactic application that allows students to experiment with networking concepts through a virtual laboratory. It provides simulation, visualization, authoring, assessment, and collaboration capabilities to facilitate the teaching and learning of complex networking technologies by visually simulating virtual networking environments (Frezzo et al., 2009). With Packet Tracer, students can easily build their network

<sup>8</sup> <http://www.vmware.com/products/player>

topologies in a visual way by dragging, placing, connecting, and clustering virtual network devices such as hubs, switches, routers, workstations, and servers. Switches and routers can be configured by students just by double-clicking their icon and entering the commands (in the same way they will enter them in real devices) in the CLI tab of the window that will appear. Packet Tracer has two operational modes: real-time and simulation mode. The real-time mode simulates a real environment, with the same speed of the simulated networks and protocols, in a similar way of real situations. In simulation mode, Packet Tracer displays the actual data exchange between devices. Each packet, or frame, that carries some data is displayed as a small envelop moving on links between devices. Students can set a filter to limit the study to a particular protocol such as OSPF and can see some information of the OSPF packets (Hello, Database Description, Link-State Request, Link-State Update, and Link-State Acknowledgment) based on the different layers of the OSI model, similarly to using a packet analyzer (sniffer) in a real network. However, the information presented in the simulation mode is simplified and may be sufficient only for introductory courses. In fact, Packet Tracer only shows the details of the Data Link (Layer 2) and Network (Layer 3) layers. For intermediate and advanced courses of OSPF, students need much more details about the information exchanged by the routers. Another important issue with Packet Tracer is that it is not possible to create interactions between a virtual and a real router.

Dynamips is a free open-source emulator for Cisco routers than run on traditional PC with Windows, Linux, or MacOS X. It can emulate Cisco 1700 series, 2600 series (2610 to 2650XM, and 2691), 3600 series (3620, 3640, and 3660), 3700 series (3725 and 3745), and 7200. In other words, Dynamips allows students to create virtual routers that run a real Cisco Internetwork Operating System (IOS) by using the PC resources. Since the IOS is a commercial product, students will have to legally get an IOS copy to use Dynamips. GNS3<sup>9</sup> is a front-end for Dynamips, that is, it is a graphical application that allows users to visually create their network topology based on Cisco routers just by dragging and clicking as they do in Packet Tracer. Since Dynamips runs true IOSs, it supports all the OSPF commands that are implemented in the IOS. That is, it does not have the limitations of Packet Tracer; however, users are restricted to smaller topologies due to the resources needed by each virtual router. With Dynamips, virtual routers can interact with real routers allowing students to expand their testbed to a bigger topology. Another important feature for the training of students that offers Dynamips is the capture of network traffic. To do so, from GNS3, students just have to right click the link where they want to capture and choose the “Capture” item in the context menu. Immediately, Wireshark (Sanders, 2007) will appear and students will get a copy of all the OSPF packets (Hello, Database Description, Link-State Request, Link-State Update, and Link-State Acknowledgment) sent by routers in the link. Dynamips/GNS3 seems to be a great solution for the teaching and learning of OSPF, however many educational entities will not be able to afford the cost of the IOS. Last, but not least, Dynamips/GNS3 lacks of a dynamic graphical network map where students can see the OSPF topology knows by a specific router.

There are several research projects which use virtualization technology as a basic tool for the realization of virtual network laboratories. They are aimed to be used in educational environment and to help in testing network applications over complex testbeds made of several nodes and networks, inside one Linux or BSD machine, without involving the investment and management complexity needed to create them using real equipment. Integrated Multiprotocol Network Emulator/Simulator (IMUNES) (Puljiz and Mikuc, 2006) is based on a lightweight virtual machine concept. It works on a modified FreeBSD kernel and enables emulated nodes to use standard Unix applications. Virtual Network User Mode Linux (VNUML) (Galán et al., 2004) is an open-source general purpose virtualization tool designed to quickly define and test complex network simulation scenarios based on the User Mode Linux (UML) virtualization software. Netkit<sup>10</sup> is an environment initially aimed for experimental usage in the area of computer networks. Netkit also uses UML and the topology is described through NetML, a XML-based language. Marionnet (Loddo and Saiu, 2008) is a virtual network laboratory build on top of UML. It allows users to define, configure, and run complex computer networks (computers, hubs, switches, and routers) on a single host machine. Individual virtual devices can be dynamically created, destroyed, or modified while the rest of the virtual network is running, providing many opportunities for experimentation

---

<sup>9</sup> <http://www.gns3.net>

<sup>10</sup> <http://www.netkit.org>



without any need for clumsy hardware setup. All these projects used Zebra, XORP<sup>11</sup> (Extensible Open Router Platform), or Quagga for the dynamic routing software, and therefore support the common routing protocols (RIP, OSPF, IS-IS, and BGP). Unlike Easy-OSPF, some have a poor graphical interface and have no special support for OSPF, i.e., none of these tools allows the users to see a dynamic view of the OSPF network on a particular router.

Other approaches to transfer OSPF's knowledge are based on network simulators such as ns-2, ns-3, OMNet++, QualNet, and OPNET. A comparative study of network simulation tools can be found in (Gamess and Veracochea, 2010). In (Zengin and Sarjoughian, 2010), the DEVS-Suite general-purpose simulator was extended to support the modeling of network protocols. A model library for the OSPF protocol was developed such that the emphasis was placed on education to capture the basic principles of network protocols, using sound modeling and simulation principles instead of supporting highly detailed network protocol simulations. However, this approach is not really suitable for novice users since it requires intermediate understanding of network simulation tools and the Java programming language. Additionally, in our opinion, this methodology is not closed to a real and practical environment.

## 6. CONCLUSION AND FUTURE WORKS

In this paper, we presented a visual teaching and learning environment for the OSPF routing protocol (Easy-OSPF). As shown in Section 4, Easy-OSPF can dramatically help students in acquiring advanced routing protocol concepts, by setting up a testbed and interacting with an integrated visual environment.

On February 3, 2011, the central pool of IPv4 addresses administrated by the Internet Assigned Numbers Authority (IANA) has been depleted (Internet Assigned Numbers Authority, 2011). According to the global policy, the last available IPv4 address blocks (102/8, 103/8, 104/8, 179/8, and 185/8) were allocated, one to each of the five Regional Internet Registries (RIR) around the world. The exhaustion of IPv4 addresses has triggered the unavoidable process of transitioning to the new Internet protocol, also known as IPv6 (Internet Protocol version 6) (Davies, 2008). IPv6 fixes a number of problems in IPv4, such as the limited number of available IPv4 addresses. IPv6 has 128-bit addresses while IPv4 has 32-bit addresses. IPv6 also adds many improvements to IPv4 in areas such as quality of service, routing, and network auto-configuration. For a successful transition to the new protocol, a huge number of IPv6 experts are needed. Hence, some works have been done for the teaching and learning of advanced IPv6's concepts (Abadi et al., 2010, Chiu and Gamess, 2010). However, there are still a lot of efforts to be done in this area to achieve the goal of having sufficient network administrators with IPv6 advanced skills. Thus, as future works, we are planning to integrate IPv6 in Easy-OSPF according to the specification of OSPFv3 (Coltun et al., 2008).

Our long-term objective is to develop an integrated visual environment for the teaching and learning of routing protocols. Our plan is to offer an easy-to-use didactic application which will included all the common routing protocols, such as RIP, OSPF, IS-IS, EIGRP, and BGP. To fulfill the objective, we already had developed Easy-EIGRP (Expósito et al., 2010a, Expósito et al., 2010b, Trujillo et al., 2010), and it had been used successfully in several courses in our Computer Science Department. Easy-OSPF and Easy-EIGRP have the same GUI for a better usability in the final integrated application. The idea is to present a similar graphical interface for all the routing protocols to minimize the learning curve of the whole integrated application. In the upcoming months, we also plan to proceed with the development of the missing protocols.

## REFERENCES

- Abadi, G., Jimenez, F., and Gamess, E. (2010). "wradvs: A Didactic Server for IPv6 Stateless Autoconfiguration". *Technological Developments in Networking, Education and Automation*. Springer, DOI 10.1007/978-90-481-9151-2. pp 443-448.
- Baker, B. (1995). "Requirements for IP Version 4 Routers". RFC 1812.

---

<sup>11</sup> <http://www.xorp.org>

- Chiu, S. and Gamess, E. (2010). "A Free and Didactic Implementation of the SEND Protocol for IPv6". Machine Learning and Systems Engineering, Lecture Notes in Electrical Engineering 68, DOI 10.1007/978-90-481-9419-3\_35, Springer Science. pp 451-463.
- Coltun, R., Ferguson, D., Moy, J., and Lindem, A. (2008). "OSPF for IPv6". RFC 5340.
- Davies, J. (2008). "Understanding IPv6". Microsoft Press. Second Edition.
- Doyle, J. and Carroll, J. (2005). "Routing TCP/IP, Volume 1". Cisco Press. Second Edition.
- Expósito, J., Trujillo, V., and Gamess, E. (2010a). "Easy-EIGRP: A Didactic Application for Teaching and Learning of the Enhanced Interior Gateway Routing Protocol". The sixth International Conference on Networking and Services (ICNS 2010), Cancun, Mexico.
- Expósito, J., Trujillo, V., and Gamess, E. (2010b). "Using Visual Educational Tools for the Teaching and Learning of EIGRP". The 2010 International Conference on Computer Science and Applications (ICCSA'10), San Francisco, California, USA.
- Frezzo, D., Behrens, J., Mislevy, R., West, P., and DiCerbo, K. (2009). "Psychometric and Evidentiary Approaches to Simulation Assessment in Packet Tracer Software". The Fifth International Conference on Networking and Services (ICNS 2009). Valencia, Spain.
- Galán, F., Fernández, D., Rúa, J., Walid, O., and Miguel, T. (2004). "A Virtualization Tool in Computer Network Laboratories". 5th International Conference on Information Technology Based Higher Education and Training (ITHET'04). Istanbul, Turkey.
- Galitz, O. (2007). "The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques". Third Edition. Wiley Publishing.
- Gamess, E. and Veracoechea, C. (2010). "A Comparative Analysis of Network Simulation Tools". The 2010 International Conference on Modeling, Simulation and Visualization Methods (MSV'10). Las Vegas, Nevada, USA.
- Internet Assigned Numbers Authority. (2011). "IPv4 Address Space Registry". Available online: <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.txt>.
- Lammle, T. (2011). "CCNA Cisco Certified Network Associate Study Guide". Sybex. Seventh Edition.
- Loddo, J.-V. and Saiu, L. (2008). "Marionnet: A Virtual Network Laboratory and Simulation Tool". SimulationWorks. Marseille, France.
- Manning, W. (2010). "CCIE Cisco Certified Internetwork Expert Routing and Switching". Emereo Pty Ltd.
- Moy, J. (1998). "OSPF Version 2". RFC 2328.
- Odom, W. (2010). "CCNP ROUTE 642-902 Official Certification Guide". Cisco Press.
- Puljiz, Z. and Mikuc, M. (2006). "IMUNES Based Distributed Network Emulator". The International Conference on Software in Telecommunications and Computer Networks (SoftCOM 2006).
- Sanders, C. (2007). "Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems". No Starch Press. First Edition.
- Skeet, J. (2010). "C# in Depth". Manning Publications. Second Edition.
- Trujillo, V., Expósito, J., and Gamess, E. (2010). "An Alternative Way of Teaching the Advanced Concepts of the Diffusing Update Algorithm for EIGRP". The 2010 International Conference on Computer Science and Applications (ICCSA'10). San Francisco, California, USA.
- Zengin, A. and Sarjoughian, H. (2010). "DEVS-Suite Simulator: A Tool Teaching Network Protocols". The 2010 Winter Simulation Conference. Baltimore, Maryland, USA.

## ACKNOWLEDGMENT

We want to thank the CNTI (Centro Nacional de Tecnologías de Información) which partially supported this research.

## *Authorization and Disclaimer*

*Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.*